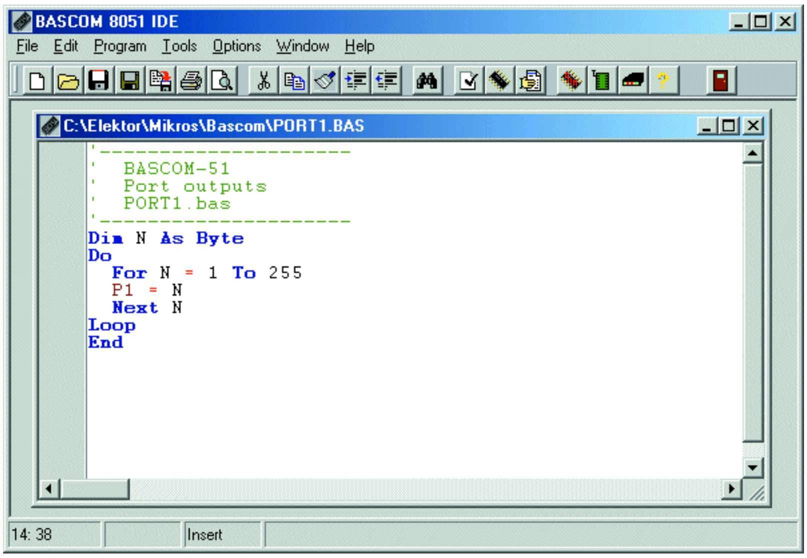
BASCOM-51

a new BASIC compiler for the Elektor 89S8252 Flash Board

By B. Kainka

Till now, the BASIC-52 interpreter was the only choice if you want to run BASIC on the extremely popular Elektor Flash Microcontroller Board (described in the December 2001 issue). Fans of the BASIC programming language will like to hear that there is now a powerful alternative called BASCOM-51.



C:\Elektor\Mikros\Bascom\PORT1.BAS

```
'--------------------
'  BASCOM-51
'  Port outputs
'  PORT1.bas
'--------------------
Dim N As Byte
Do
  For N = 1 To 255
  P1 = N
  Next N
Loop
End
```

14: 38 Insert

Figure 1. The programming environment.

BASCOM was originally developed by Mark Alberts and his Dutch company MCS-Electronics (www.mcselec.com) for the AVR series of microcontrollers from Atmel. Later, a spin-off called BASCOM-LT was marketed specially for the smaller controller type 89C2051. This eventually led to BASCOM-51 being developed and sold for the beefed up derivates of the 8051. The MCS-Electronics website has available a free download of the BASCOM-51 demo version. This version allows programs with a size of up to 2 kBytes to be developed. This may not seem a lot but don't be fooled because the compiler is very effective at generating code! The free demo version was tested in combination with the *Elektor Electronics* 89S8252 Flash Microcontroller Board (December 2001) with excellent results.

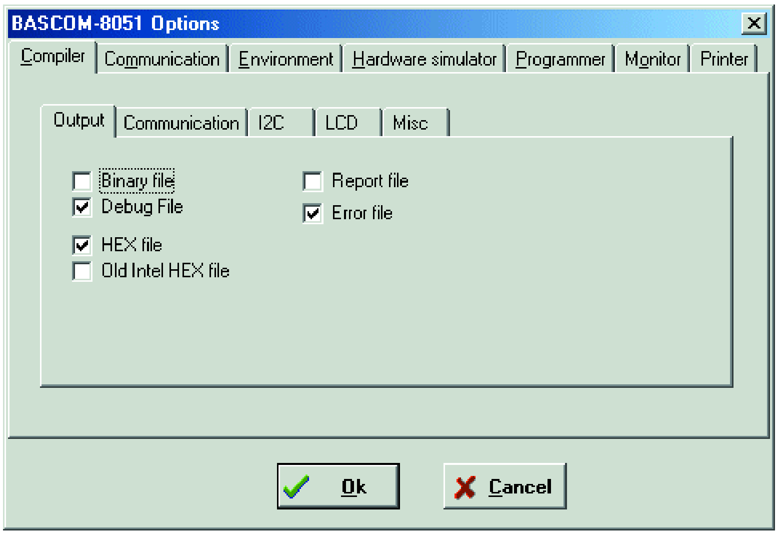
The compiler is marked by a large number of important settings. Under Options/Compiler/Output you can select the desired output format. Here, Hexfile is the format to go for because it can be read by a program like MicroFlash or AtmelISP. From there, executable code may be copied into the controller. The Debug file is essential for the built-in Simulator. The Error file is used to draw your attention to errors that occurred during the compilation process.

Another important setting is found under Options/Compiler/Output/Misc. which is where you can select the Register file, in this case, the file `89s8252.dat`. Once this has been selected, the compiler will recognize all special function registers (SFRs) inside the controller.

A first test may be performed using a small program for fast bit outputting via the ports. The first thing you'll notice is the absence of line numbers, which is evidence of structured BASIC being implemented. Instead of `GOTO` and line numbers, instructions like `Do` and `While` are employed. In this way, BASCOM follows a trend in that higher-level programming languages seem to develop increasing 'common ground'.

The second pleasant thing to notice is that all variables have to be declared with their desired type. This allows you, for instance, to work with individual bytes and so approach the results offered by an assembler. Moreover, you can explicitly state whether or not variables are allowed in the internal RAM. BASCOM-51 can make do without external memory, which is not self-evident for a compiler.

```
'-----------
'  BASCOM-51
'  Port outputs
'  PORT1.bas
'-----------
Dim N As Byte
Do
  For N = 1 To 255
  P1 = N
  Next N
Loop
End
```



☐ Binary file
☑ Debug File
☑ HEX file
☐ Old Intel HEX file
☐ Report file
☑ Error file

Figure 2. Compiler settings.

As with any new higher programming language, the main question is: 'so how fast is the code'? The answer is given by an oscilloscope hooked up to port line P1.1. The result: only 30 µs for each new output action. Admittedly, assembler is about 10 times faster at just over 3 µs, but then the READS-51 C Compiler requires 125 µs for the same task, not even mentioning BASIC-52, which takes no less than 2,500 µs. This makes BASCOM-51 a clear winner in its class of compilers and interpreters. Compared with BASIC-52, there's another advantage in that the code generated by BASCOM-51 can be made self-starting without problems.

The new compiler offers different serial interfaces. The commands `PRINT` and `INPUT` convey data via the UART inside the microcontroller. The baudrate may be found among the settings under Options/Compiler/Communication.

```
'———————-
'  BASCOM-51
'  Port inputs and outputs
'  RS232.bas
'———————-
Dim N As Byte

While 1 = 1
  Input N
  P1 = N
  N = P1
  Print N
Wend
```

Besides the hardware UART, you have a software UART so that a number of interfaces are available, also when using the Elektor 89S8252 Flash Microcontroller Board. The next program listing demonstrates the use of the software UART. For a first test, the same pins are used as with the processor's on-chip UART. For the software UART, individual bytes are processed using the commands `GET` and `PUT`.

```
'——————————
' BASCOM-51
' Port inputs and outputs
' RS232_2.bas software uart
'——————————
Dim N As Byte

Open “com3.1:9600” For Output
As #1
Open “com3.0:9600” For Input As #2

While 1 = 1
  Get #2 , N
  P1 = N
  N = P1
  Put #1 , N
Wend

Close #1
Close #2
```

BASCOM-51 not only processes byte-ports, but also individual bits. The following example program divides the signal frequency applied to port P1.0 and outputs it frequency-halved on port P1.1. This is done by toggling the P1.1 bit on every falling edge of the input signal. In this way, the operation of the program may be compared to that of a binary counter. The maximum input frequency will be about 50 Hz.

```
'——————————
'  BASCOM-51
'  Port input freq. divider
'  Toggle.bas
'——————————

While 1 = 1
  While P1.0 = 1
  Wend
  P1.1 = Not P1.1
  While P1.0 = 0
  Wend
Wend
```

If necessary, chunks of assembler code may be inserted. The example below demonstrates how fast pulses spaced 10 ms apart are generated on port P1. The longer delay is of course generated in BASIC, the time-critical parts, in assembler.

```
'——————————
' BASCOM-51
' Ass. example Assem.bas
' mixed assembler and Basic
'——————————
Print “pulses”

While 1 = 1
$asm
   mov A , #00
   mov R1,#100
L1:
   mov P1,A
   inc a
   djnz R1,L1
```

Table 1. Compiler/Assembler comparisons

language	**Memory req.**	**Loop time**	**Autostart**
Basic-52	8 K ROM, RAM	2500 µs	only with EEPROM
READS51 C	>4 K ROM, RAM	125 µs	yes
Assembler	< 1 K ROM	3 µs	yes
BASCOM-51	< 2 K ROM	30 µs	yes

```
$end Asm
Waitms 10
Wend
End
```

BASCOM-51 has far more to offer than can be described in this short article. For example, a number of different interfaces for special peripherals are supported: LC displays in 4-bit and 8-bit mode, I^2C devices, the 1-Wire bus, and the software interface discussed above. Many tasks are considerably simplified in this way. We would therefore suggest that every user of 8051 or AVR microcontrollers takes a good look at BASCOM-51.

(021003-1)

BASCOM distributors in the UK

Kanda Systems Ltd.
Units 17/18
Glanyrafon Enterprise Park
Aberystwyth
Ceredigion SY23 3JQ.
Tel. (+44) (0)1970 621030
Email: *sales@kanda.com*
Website: *www.kanda.com*

Quasar Electronics
Mr. Simon Neil
Unit 14 Sunningdale
Bishops Stortford
Herts CM23 2PA.
Tel. (+44) (0)1279 306504
Fax: (+44) (0)870 7064222
Email: *simon@quasarelectronics.com*
Website: *www.quasarelectronics.com*

A list of worldwide distributors for BASCOM-51 may be found on the 'Resellers' page at *www.mcselec.com*